A Largish Sum-of-Squares Implies Circuit Hardness and Derandomization

Pranjal Dutta (CMI & IIT Kanpur)  
Nitin Saxena (IIT Kanpur)  
Thomas Thierauf (Aalen University)

22\textsuperscript{nd} September, 2020  
tMeet @CSE, IIT Madras (Online)
# Table of contents

1. Introduction: Sum-of-squares (SOS)

2. Basic Algebraic Complexity

3. SOS-hardness and VP vs. VNP

4. Sum-of-cubes (SOC) model and Blackbox-PIT

5. Conclusion
Introduction: Sum-of-squares (SOS)
An $n$-variate polynomial $f(x) \in F[x]$ over a field $F$ is computed as a sum-of-squares (SOS) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,$$

for some top-fanin $s$, where $f_i(x) \in F[x]$ and $c_i \in F$.

$|f|_0$ denotes sparsity of $f$. Eg. $f(x) = 2x + 2 = (x + 3/2)^2 - (x + 1/2)^2$. Size of $f$ in this SOS representation is $2 + 2 = 4$.

Denote the minimal size by support-sum $S_F(f)$.

Note. SOS is a complete model if $\text{char}(F) \neq 2$, as $f = (f + 1/2)^2 - (f - 1/2)^2$.

Trivially, $S_F(f) \leq 2 \cdot (|f|_0 + 1)$, for any $f \in F[x]$. 
An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a *sum-of-squares* (SOS) if

\[
f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,
\]

for some *top-fanin* $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$. 
Sum-of-squares (SOS) Representation

An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a *sum-of-squares* (SOS) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,$$

for some *top-fanin* $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

- **Size** of $f$ in Eqn. (1) is no. of monomials $= \sum_{i \in [s]} |f_i|_0$. $|f|_0$ denotes sparsity of $f$. 

An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a *sum-of-squares* (SOS) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,$$

for some *top-fanin* $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

- **Size** of $f$ in Eqn. (1) is no. of monomials $= \sum_{i \in [s]} |f_i|_0$. $|f|_0$ denotes sparsity of $f$.

- Eg. $f(x) := 2x + 2 = (x + 3/2)^2 - (x + 1/2)^2$. Size of $f$ in this SOS representation is $2 + 2 = 4$. 
An \( n \)-variate polynomial \( f(x) \in \mathbb{F}[x] \) over a field \( \mathbb{F} \) is computed as a \textit{sum-of-squares} (SOS) if

\[
f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,
\]

for some \textit{top-fanin} \( s \), where \( f_i(x) \in \mathbb{F}[x] \) and \( c_i \in \mathbb{F} \).

- **Size of** \( f \) in Eqn. (1) is no. of monomials = \( \sum_{i \in [s]} |f_i|_0 \). \( |f|_0 \) denotes sparsity of \( f \).

- Eg. \( f(x) := 2x + 2 = (x + 3/2)^2 - (x + 1/2)^2 \). Size of \( f \) in this SOS representation is 2 + 2 = 4.

- Denote the \textit{minimal size} by \textit{support-sum} \( S_{\mathbb{F}}(f) \).
An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a *sum-of-squares* (SOS) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^2,$$

for some *top-fanin* $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

- **Size** of $f$ in Eqn. (1) is no. of monomials $= \sum_{i \in [s]} |f_i|_0$. 
  
  Eg. $f(x) := 2x + 2 = (x + 3/2)^2 - (x + 1/2)^2$. Size of $f$ in this SOS representation is $2 + 2 = 4$.

- Denote the *minimal size* by support-sum $S_{\mathbb{F}}(f)$.

**Note.** SOS is a *complete* model if char($\mathbb{F}$) $\not= 2$, as $f = \left( \frac{f+1}{2} \right)^2 - \left( \frac{f-1}{2} \right)^2$. 

An $n$-variate polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ over a field $\mathbb{F}$ is computed as a **sum-of-squares** (SOS) if

$$f(\mathbf{x}) = \sum_{i=1}^{s} c_i \cdot f_i(\mathbf{x})^2,$$

for some *top-fanin* $s$, where $f_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ and $c_i \in \mathbb{F}$.

- **Size** of $f$ in Eqn. (1) is no. of monomials = $\sum_{i \in [s]} |f_i|_0$.  
  - Eg. $f(x) := 2x + 2 = (x + 3/2)^2 - (x + 1/2)^2$. Size of $f$ in this SOS representation is $2 + 2 = 4$.

- Denote the *minimal size* by support-sum $S_{\mathbb{F}}(f)$.

**Note.** SOS is a *complete* model if $\text{char}(\mathbb{F}) \neq 2$, as $f = \left(\frac{f+1}{2}\right)^2 - \left(\frac{f-1}{2}\right)^2$.  
  - Trivially, $S_{\mathbb{F}}(f) \leq 2 \cdot (|f|_0 + 1)$, for any $f \in \mathbb{F}[\mathbf{x}]$. 
For simplicity, consider univariate SOS representations ($n = 1$).
For simplicity, consider univariate SOS representations \((n = 1)\).

For any \(\text{char}(\mathbb{F}) \neq 2\) field \(\mathbb{F}\):

\[
|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2|f|_0 + 2. \tag{2}
\]

Lower bound by counting monomials:
For simplicity, consider univariate SOS representations \((n = 1)\).

For any char \((\mathbb{F}) \neq 2\) field \(\mathbb{F}\):

\[
|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2|f|_0 + 2. \tag{2}
\]

Lower bound by counting monomials:

- Suppose \(f = \sum_{i=1}^{s} c_i \cdot f_i^2\). Assume, \(|f_i|_0 = t_i\).

- Note, \(|f_i^2|_0 \leq t_i^2\), for each \(i \in [s]\).

- \(\sum_{i=1}^{s} t_i^2 \geq |f|_0 \implies \sum_{i=1}^{s} t_i \geq |f|_0^{1/2}\).
Upper bound and lower bound: What to expect

- For simplicity, consider univariate SOS representations \((n = 1)\).

- For any \(\text{char}(\mathbb{F}) \neq 2\) field \(\mathbb{F} \):

\[
|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2|f|_0 + 2.
\]

(2)

Lower bound by counting monomials:

- Suppose \(f = \sum_{i=1}^{s} c_i \cdot f_i^2\). Assume, \(|f_i|_0 = t_i\).

- Note, \(|f_i^2|_0 \leq t_i^2\), for each \(i \in [s]\).

- \(\sum_{i=1}^{s} t_i^2 \geq |f|_0 \implies \sum_{i=1}^{s} t_i \geq |f|_0^{1/2}\).

- If \(|f|_0 \approx d\), then \(\Omega(d^{1/2}) \leq S_{\mathbb{F}}(f) \leq O(d)\).
For simplicity, consider univariate SOS representations \((n = 1)\).

For any char\((\mathbb{F}) \neq 2\) field \(\mathbb{F}\):

\[
\left| f \right|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2 \left| f \right|_0 + 2.
\] (2)

Lower bound by counting monomials:

- Suppose \(f = \sum_{i=1}^{s} c_i \cdot f_i^2\). Assume, \(\left| f_i \right|_0 = t_i\).
- Note, \(\left| f_i^2 \right|_0 \leq t_i^2\), for each \(i \in [s]\).
- \(\sum_{i=1}^{s} t_i^2 \geq \left| f \right|_0 \implies \sum_{i=1}^{s} t_i \geq \left| f \right|_0^{1/2}\).

If \(\left| f \right|_0 \approx d\), then \(\Omega(d^{1/2}) \leq S_{\mathbb{F}}(f) \leq O(d)\).

Does there exist \(d\)-degree polynomial \(f(x)\) such that \(S_{\mathbb{F}}(f) \geq \Omega(d)\)?
Upper bound and lower bound: What to expect

- For simplicity, consider univariate SOS representations \( n = 1 \).

- For any char(\( \mathbb{F} \)) \( \neq 2 \) field \( \mathbb{F} \):

\[
|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2|f|_0 + 2.
\] (2)

Lower bound by counting monomials:

- Suppose \( f = \sum_{i=1}^{s} c_i \cdot f_i^2 \). Assume, \( |f_i|_0 = t_i \).

- Note, \( |f_i^2|_0 \leq t_i^2 \), for each \( i \in [s] \).

- \( \sum_{i=1}^{s} t_i^2 \geq |f|_0 \implies \sum_{i=1}^{s} t_i \geq |f|_0^{1/2} \).

- If \( |f|_0 \approx d \), then \( \Omega(d^{1/2}) \leq S_{\mathbb{F}}(f) \leq O(d) \).

- Does there exist \( d \)-degree polynomial \( f(x) \) such that \( S_{\mathbb{F}}(f) \geq \Omega(d) \)?
  - True for “most" polynomials \( f \), by dimension-argument.
For simplicity, consider univariate SOS representations \((n = 1)\).

For any char\((\mathbb{F}) \neq 2\) field \(\mathbb{F}\):

\[
|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq 2|f|_0 + 2. \tag{2}
\]

Lower bound by counting monomials:

- Suppose \(f = \sum_{i=1}^s c_i \cdot f_i^2\). Assume, \(|f_i|_0 = t_i\).
- Note, \(|f_i^2|_0 \leq t_i^2\), for each \(i \in [s]\).
- \(\sum_{i=1}^s t_i^2 \geq |f|_0 \implies \sum_{i=1}^s t_i \geq |f|_0^{1/2}\).

If \(|f|_0 \approx d\), then \(\Omega(d^{1/2}) \leq S_{\mathbb{F}}(f) \leq O(d)\).

Does there exist \(d\)-degree polynomial \(f(x)\) such that \(S_{\mathbb{F}}(f) \geq \Omega(d)\)?

- True for “most” polynomials \(f\), by dimension-argument.
- Assume, \(\mathbb{F} = \mathbb{C}\).
Open Problem. Find an explicit univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree $d$ such that $S(f) \geq \Omega(d/\log d)$, where $f(x) = \sum_{i=0}^d 2^i x^i$, using [Strassen’74]. But, it is non-explicit.

To be of any help in complexity theory, polynomials need to be explicit. We would work with several definitions of explicitness. E.g. $(x+1)^d$ is ‘explicit’.

Overall Goal (informally): Show that solving Open Problem implies $\text{VP} \neq \text{VNP}$ (and $\text{PIT} \in \text{SUBEXP}$).
Open Problem. Find an explicit univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree $d$ such that $S(f) \geq \omega(d^{1/2})$. 
☐ **Open Problem.** Find an *explicit* univariate polynomial \( f(x) \in \mathbb{C}[x] \) of degree \( d \) such that \( S(f) \geq \omega(d^{1/2}) \).

➢ \( S(f) \geq \Omega(d / \log d) \), where \( f(x) = \sum_{i=0}^{d} 2^{2^i} x^i \), using [Strassen’74].
- **Open Problem.** Find an *explicit* univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree $d$ such that $S(f) \geq \omega(d^{1/2})$.

  - $S(f) \geq \Omega\left(\frac{d}{\log d}\right)$, where $f(x) = \sum_{i=0}^{d} 2^{2^i} x^i$, using [Strassen’74]. But, it is *non-explicit*.
Open Problem. Find an explicit univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree $d$ such that $S(f) \geq \omega(d^{1/2})$.

- $S(f) \geq \Omega(d/\log d)$, where $f(x) = \sum_{i=0}^{d} 2^{2^i} x^i$, using [Strassen’74]. But, it is non-explicit.

- To be of any help in complexity theory, polynomials need to be explicit. We would work with several definitions of explicitness.

- Eg. $(x + 1)^d$ is ‘explicit’.
Open Problem. Find an explicit univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree $d$ such that $S(f) \geq \omega(d^{1/2})$.

- $S(f) \geq \Omega(d/\log d)$, where $f(x) = \sum_{i=0}^{d} 2^{2i} x^i$, using [Strassen’74]. But, it is non-explicit.

- To be of any help in complexity theory, polynomials need to be explicit. We would work with several definitions of explicitness.

- Eg. $(x + 1)^d$ is ‘explicit’.

Overall Goal (informally): Show that solving Open Problem implies $VP \neq VNP$ (and PIT $\in$ SUBEXP).
SOS Representation – History

(1770) Lagrange's 4-squares Theorem: Integer as sum of 4-squares. Inspired generations of mathematicians [Ramanujan'17].

(1900) Hilbert's 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?

Note: $c_i = 1$.

(1990s) SOS constraints appear in convex optimization. Lasserre hierarchy of relaxations in SDP (based on deg). Several applications in approximation, optimization and control theory [Reznick'78, Laurent'09, Barak-Moitra'16].
(1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.
(1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.

- Inspired generations of mathematicians [Ramanujan’17].
(1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.

- Inspired generations of mathematicians [Ramanujan’17].
- Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

(1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions.

Note: $c_i = 1$.

(1990s) SOS constraints appear in convex optimization.

Lasserre hierarchy of relaxations in SDP (based on deg).

Several applications in approximation, optimization and control theory [Reznick’78, Laurent’09, Barak-Moitra’16].
(1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.

- Inspired generations of mathematicians [Ramanujan’17].
- Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

(1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?
(1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.

- Inspired generations of mathematicians [Ramanujan’17].
- Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

(1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?

- Note: $c_i = 1$. 

SOS Representation – History

- (1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.
  - Inspired generations of mathematicians [Ramanujan’17].
  - Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

- (1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?
  - Note: $c_i = 1$.

- (1990s) SOS constraints appear in convex optimization.
SOS Representation – History

- (1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.
  - Inspired generations of mathematicians [Ramanujan’17].
  - Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

- (1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?
  - Note: \(c_i = 1\).

- (1990s) SOS constraints appear in convex optimization.
  - Lasserre hierarchy of relaxations in SDP (based on deg).
SOS Representation – History

- **(1770) Lagrange’s 4-squares Theorem:** Integer as sum of 4-squares.
  - Inspired generations of mathematicians [Ramanujan’17].
  - Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

- **(1900) Hilbert’s 17th problem:** Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?
  - Note: $c_i = 1$.

- **(1990s) SOS constraints** appear in convex optimization.
  - *Lasserre hierarchy* of relaxations in SDP (based on deg).
  - Several applications in approximation, optimization and control theory [Reznick’78, Laurent’09, Barak-Moitra’16].
SOS Representation – History

- (1770) Lagrange’s 4-squares Theorem: Integer as sum of 4-squares.
  - Inspired generations of mathematicians [Ramanujan’17].
  - Pythagorean triples, Fermat’s 2-squares, Legendre’s 3-squares.

- (1900) Hilbert’s 17th problem: Asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions?
  - Note: $c_i = 1$.

- (1990s) SOS constraints appear in convex optimization.
  - Lasserre hierarchy of relaxations in SDP (based on deg).
  - Several applications in approximation, optimization and control theory [Reznick’78, Laurent’09, Barak-Moitra’16].
Basic Algebraic Complexity
Algebraic Circuits

\[ f(x) = x_1 \times x_2 \times x_3 + 1 \]
Algebraic Circuits

\[
f(x) = x_1 \times x_2 \times x_3 + 1
\]
Algebraic Circuits

Size $= \text{number of nodes} + \text{edges}$
Valiant’s Hypothesis [Valiant’79]: Symbolic perm$_n$ requires $n^{\omega(1)}$-size circuit.
Valiant's Hypothesis [Valiant’79]: Symbolic perm\(_n\) requires \(n^{\omega(1)}\)-size circuit.

An equivalent statement: Prove \(\text{VP} \neq \text{VNP}\).
Valiant’s Hypothesis [Valiant’79]: Symbolic perm$_n$ requires $n^{\omega(1)}$-size circuit.

An equivalent statement: Prove $\text{VP} \neq \text{VNP}$.

VP: A family $(f_n)_n \in \text{VP}$ (over $F$) if $f_n$ is a poly$(n)$-variate polynomial, of degree poly$(n)$ over $F$, computed by poly$(n)$-size circuit.
Valiant’s Hypothesis [Valiant’79]: Symbolic perm$_n$ requires $n^{\omega(1)}$-size circuit. An equivalent statement: Prove $\text{VP} \neq \text{VNP}$.

- **VP**: A family $(f_n)_n \in \text{VP}$ (over $\mathbb{F}$) if $f_n$ is a $\text{poly}(n)$-variate polynomial, of degree $\text{poly}(n)$ over $\mathbb{F}$, computed by $\text{poly}(n)$-size circuit.

- **VNP**: A family $(f_n)_n \in \text{VNP}$ (over $\mathbb{F}$) if $\exists (g_n)_n \in \text{VP}$ & $t(n) = \text{poly}(n)$:
- **Valiant’s Hypothesis** [Valiant’79]: Symbolic perm$_n$ requires $n^{\omega(1)}$-size circuit. An equivalent statement: Prove $\text{VP} \neq \text{VNP}$.

- **VP**: A family $(f_n)_n \in \text{VP}$ (over $F$) if $f_n$ is a poly$(n)$-variate polynomial, of degree $\text{poly}(n)$ over $F$, computed by poly$(n)$-size circuit.

- **VNP**: A family $(f_n)_n \in \text{VNP}$ (over $F$) if $\exists (g_n)_n \in \text{VP}$ & $t(n) = \text{poly}(n)$:

$$f_n(x) = \sum_{w \in \{0,1\}^{t(n)}} g_n(x, w).$$
Polynomial Identity Testing (PIT): Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)
If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq F$ is finite, then
$$\text{Prob}_{a \in S}[P(a) = 0] \leq \frac{d}{|S|}.$$ 

The above lemma puts PIT $\in \text{RP}$.

Hardness-to-randomness (Kabanets-Impagliazzo'04)
$	ext{VP} \neq \text{VNP} = \Rightarrow \text{PIT} \in \text{SUBEXP}$.

$\text{VNP}$ is exponentially harder than $\text{VP} = \Rightarrow \text{PIT} \in \text{QP}$.

Efficient PIT $\Rightarrow \text{VP} \neq \text{VNP}$.

Explicitness is important.
Polynomial Identity Testing (PIT): Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

- Blackbox-PIT asks for an algorithm to test the zeroness of a given algebraic circuit via mere *query access*. 

Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq F$ is finite, then

$$\Pr_{a \in S} [P(a) = 0] \leq \frac{d}{|S|}.$$

The above lemma puts PIT $\in \text{RP}$.

Hardness-to-randomness (Kabanets-Impagliazzo'04)

$\text{VP} \neq \text{VNP} \Rightarrow \text{PIT} \in \text{SUBEXP}$.

- $\text{VNP}$ is exponentially harder than $\text{VP} = \Rightarrow \text{PIT} \in \text{QP}$.

Efficient PIT $\Rightarrow \text{VP} \neq \text{VNP}$.

Explicitness is important.
Polynomial Identity Testing (PIT): Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

> Blackbox-PIT asks for an algorithm to test the zeroness of a given algebraic circuit via mere query access.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq \mathbb{F}$ is finite, then

$$\text{Prob}_{a \in S^n} [P(a) = 0] \leq d / |S|.$$
Polynomial Identity Testing (PIT): Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

- **Blackbox-PIT** asks for an algorithm to test the zeroness of a given algebraic circuit via mere *query access*.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq \mathbb{F}$ is finite, then

$$\operatorname{Prob}_{a \in S^n} [P(a) = 0] \leq d/|S|.$$

- The above lemma puts PIT $\in \text{RP}$.
Polynomial Identity Testing (PIT): Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

- **Blackbox-PIT** asks for an algorithm to test the zeroness of a given algebraic circuit via mere *query access*.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq \mathbb{F}$ is finite, then

$$\text{Prob}_{a \in S} [P(a) = 0] \leq d/|S|.$$ 

- The above lemma puts PIT $\in$ RP.

**Hardness-to-randomness (Kabanets-Impagliazzo’04)**

$\text{VP} \neq \text{VNP} \implies \text{PIT} \in \text{SUBEXP}.$
Polynomial Identity Testing

**Polynomial Identity Testing (PIT):** Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

- **Blackbox-PIT** asks for an algorithm to test the zeroness of a given algebraic circuit via mere query access.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq \mathbb{F}$ is finite, then

$$\text{Prob}_{a \in S^n} [P(a) = 0] \leq d/|S|.$$ 

- The above lemma puts PIT $\in$ RP.

**Hardness-to-randomness (Kabanets-Impagliazzo’04)**

$\text{VP} \neq \text{VNP} \implies \text{PIT} \in \text{SUBEXP}.$

- VNP is *exponentially* harder than VP $\implies$ PIT $\in$ QP.
Polynomial Identity Testing (PIT): Given a circuit \( C \), test whether \( C \equiv 0 \) (deterministically).

- **Blackbox-PIT** asks for an algorithm to test the zeroness of a given algebraic circuit via mere *query access*.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If \( P(x) \) is a nonzero polynomial of degree \( d \), and \( S \subseteq \mathbb{F} \) is finite, then

\[
\Pr_{a \in S^n} [P(a) = 0] \leq \frac{d}{|S|}.
\]

- The above lemma puts PIT \( \in \text{RP} \).

**Hardness-to-randomness (Kabanets-Impagliazzo’04)**

\( \text{VP} \neq \text{VNP} \implies \text{PIT} \in \text{SUBEXP} \).

- VNP is *exponentially* harder than VP \( \implies \text{PIT} \in \text{QP} \).

- Efficient PIT \( \implies \text{VP} \neq \text{VNP} \).
**Polynomial Identity Testing (PIT):** Given a circuit $C$, test whether $C \equiv 0$ (deterministically).

- **Blackbox-PIT** asks for an algorithm to test the zeroness of a given algebraic circuit via mere *query access*.

**Polynomial Identity Lemma (Ore, Demillo-Lipton, Schwartz, Zippel)**

If $P(x)$ is a nonzero polynomial of degree $d$, and $S \subseteq \mathbb{F}$ is finite, then

\[
\text{Prob}_{a \in S} [P(a) = 0] \leq d/|S|.
\]

- The above lemma puts PIT $\in$ RP.

**Hardness-to-randomness (Kabanets-Impagliazzo’04)**

$\text{VP} \neq \text{VNP} \implies \text{PIT} \in \text{SUBEXP}$.

- VNP is *exponentially* harder than VP $\implies$ PIT $\in$ QP.

- Efficient PIT $\implies$ VP $\neq$ VNP. *Explicitness is important.*
**Definition (Explicit Functions).** The family \((f_d(x))_d\), where \(f_d\) is univariate degree-\(d\) polynomial, is *explicit*, if its coefficient-function \(\text{coef}_{x_i}(f_d)\) is *easy*:
**Definition (Explicit Functions).** The family $(f_d(x))_d$, where $f_d$ is univariate degree-$d$ polynomial, is *explicit*, if its coefficient-function $\text{coef}_{x_i}(f_d)$ is *easy*:

- Each coefficient can be at most $\text{poly}(d)$-bits long, and
**Definition (Explicit Functions).** The family \((f_d(x))_d\), where \(f_d\) is univariate degree-\(d\) polynomial, is *explicit*, if its coefficient-function \(\text{coef}_{x^i}(f_d)\) is *easy*:

- Each coefficient can be at most \(\text{poly}(d)\)-bits long, and
- the coefficient-function gets input \((j, i, d)\) and outputs the \(j\)-th bit of the coefficient of \(x^i\) in \(f_d\) in \(\text{poly}(\log d)\)-time.
Explicitness

**Definition (Explicit Functions).** The family \((f_d(x))_d\), where \(f_d\) is univariate degree-\(d\) polynomial, is *explicit*, if its coefficient-function \(\text{coef}_{x^i}(f_d)\) is *easy*:

- Each coefficient can be at most \(\text{poly}(d)\)-bits long, and

- the coefficient-function gets input \((j, i, d)\) and outputs the \(j\)-th bit of the coefficient of \(x^i\) in \(f_d\) in
  
  `poly(\log d)`-time.
**Definition (Explicit Functions).** The family \((f_d(x))_d\), where \(f_d\) is univariate degree-\(d\) polynomial, is *explicit*, if its coefficient-function \(\text{coef}_{x^i}(f_d)\) is *easy*:

- Each coefficient can be at most \(\text{poly}(d)\)-bits long, and
- the coefficient-function gets input \((j, i, d)\) and outputs the \(j\)-th bit of the coefficient of \(x^i\) in \(f_d\) in
  - \(\text{poly}(\log d)\)-time.
  - Or, . . . in \(\#P/\text{poly}\).
Explicitness

**Definition (Explicit Functions).** The family \( (f_d(x))_d \), where \( f_d \) is univariate degree-\( d \) polynomial, is *explicit*, if its coefficient-function \( \text{coef}_{x_i}(f_d) \) is *easy*:

- Each coefficient can be at most \( \text{poly}(d) \)-bits long, and
- the coefficient-function gets input \((j, i, d)\) and outputs the \( j \)-th bit of the coefficient of \( x^i \) in \( f_d \) in
  - \( \text{poly}(\log d) \)-time.
  - Or, \ldots in \#P/poly.
  - Or, \ldots in CH.

Requires GRH to separate VP and VNP.
Definition (Explicit Functions). The family \((f_d(x))_d\), where \(f_d\) is univariate degree-\(d\) polynomial, is explicit, if its coefficient-function \(\text{coef}_{x_i}(f_d)\) is easy:

- Each coefficient can be at most \(\text{poly}(d)\)-bits long, and

- the coefficient-function gets input \((j, i, d)\) and outputs the \(j\)-th bit of the coefficient of \(x^i\) in \(f_d\) in
  
  - \(\text{poly}(\log d)\)-time.
  
  - Or, \ldots in \#P/poly.
  
  - Or, \ldots in CH.

Requires GRH to separate VP and VNP.
Definition (SOS-hardness).

An explicit univariate polynomial family \((f_d)_{d \geq 0}\) is **SOS-hard**, if \(\text{SF}(f_d) = \Omega(d^{0.5} + Y_d)\), where \(Y_d = Y(d) = \log(\log d)/\log d\) is a sub-constant function.

Remark. Hardness examples—

\[ d^{1/2}(\log d)^{\sqrt{\log d}}, \quad d^{1/2} + 0.01. \]

There are numerous candidates for \(f_d(x)\):

- The famous Pochhammer-Wilkinson polynomial \(f_d = \prod_{i=1}^d (x - i)\).
- \(-f_d = \prod_{i=0}^d 2i^2 x^i\). \(\prod_{i=0}^d 2i x^i\) is not a candidate.
- \((-x+1)^d\). \((-x+1)^d\) has poly\((\log d)\)-size circuit.
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is \emph{SOS-hard}, if \(S_F(f_d) = \Omega(d^{0.5+\epsilon})\), where \(\epsilon := \epsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.
**Definition (SOS-hardness).** An explicit univariate polynomial family $(f_d)_d$ is \textit{SOS-hard}, if $S_F(f_d) = \Omega(d^{0.5+\varepsilon})$, where $\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)$ is a sub-constant function.

**Remark.** Hardness examples—$d^{1/2} \cdot (\log d)^{\sqrt{\log d}}$, $d^{1/2+.01}$. 
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is *SOS-hard* if \(S_F(f_d) = \Omega(d^{0.5+\varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.

**Remark.** Hardness examples—\(d^{1/2} \cdot (\log d)^{\sqrt{\log d}}, d^{1/2+.01}\).

- There are numerous candidates for \(f_d(x)\):
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is SOS-hard, if \(\mathcal{S}_\mathbb{F}(f_d) = \Omega(d^{0.5+\varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.

**Remark.** Hardness examples—\(d^{1/2} \cdot (\log d)^{\sqrt{\log \log d}}, d^{1/2+.01}\).

- There are numerous candidates for \(f_d(x)\):
  - The famous *Pochhammer-Wilkinson* polynomial \(f_d := \prod_{i=1}^d (x - i)\).
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is **SOS-hard**, if \(S_{\mathbb{F}}(f_d) = \Omega(d^{0.5+\varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.

**Remark.** Hardness examples—\(d^{1/2} \cdot (\log d)^{\sqrt{\log d}}\), \(d^{1/2+.01}\).

- There are numerous candidates for \(f_d(x)\):
  - The famous *Pochhammer-Wilkinson* polynomial \(f_d := \prod_{i=1}^d (x - i)\).
  - \(f_d := \sum_{i=0}^d 2^{i^2} x^i\).
Definition (SOS-hardness). An explicit univariate polynomial family \((f_d)_d\) is SOS-hard, if \(S(F(f_d)) = \Omega(d^{0.5 + \varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.

Remark. Hardness examples—\(d^{1/2} \cdot (\log d)^{\sqrt{\log d}}, d^{1/2+0.01}\).

- There are numerous candidates for \(f_d(x)\):
  - The famous Pochhammer-Wilkinson polynomial \(f_d := \prod_{i=1}^{d} (x - i)\).
  - \(f_d := \sum_{i=0}^{d} 2^i x^i\). However, \(\sum_{i=0}^{d} 2^i x^i\) is not a candidate.
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is **SOS-hard**, if \(S_F(f_d) = \Omega(d^{0.5+\varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log \log d}{\log d}}\right)\) is a sub-constant function.

**Remark.** Hardness examples—\(d^{1/2} \cdot (\log d)^{\sqrt{\log d}}, d^{1/2+.01}\).

- There are numerous candidates for \(f_d(x)\):
  - The famous **Pochhammer-Wilkinson** polynomial \(f_d := \prod_{i=1}^{d} (x - i)\).
  - \(f_d := \sum_{i=0}^{d} 2^i x^i\).
  - \(f_d := (x + 1)^d\).
**Definition (SOS-hardness).** An explicit univariate polynomial family \((f_d)_d\) is *SOS-hard*, if \(S_{\mathbb{F}}(f_d) = \Omega(d^{0.5+\varepsilon})\), where \(\varepsilon := \varepsilon(d) = \omega\left(\frac{\sqrt{\log \log d}}{\log d}\right)\) is a sub-constant function.

**Remark.** Hardness examples— \(d^{1/2} \cdot (\log d)^{\sqrt{\log d}}, d^{1/2+.01}\).

- There are numerous candidates for \(f_d(x)\):
  - The famous *Pochhammer-Wilkinson* polynomial \(f_d := \prod_{i=1}^{d} (x - i)\).
  - \(f_d := \sum_{i=0}^{d} 2^i x^i\). \(\sum_{i=0}^{d} 2^i x^i\) is *not* a candidate.
  - \(f_d := (x + 1)^d\). \((x + 1)^d\) has \(\text{poly}(\log d)\)-size circuit.
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- **K Agrawal-Vinay’08, .., Gupta-Kamath-Kayal-Saptharishi’13, .., Agrawal-Ghosh-Saxena’18**: Hardness for special depth-4/3 – sum-of unbounded-powers of multivariates $l(x)^{1/(d)}$.
- **Koiran’11**: Used univariate depth-4 expression of unbounded-powers; also lower bound on the top-fanin (we require SOS-size).
- SOS-size is neither top-fanin nor the “size” of the depth-4 circuits, rather it is $\#^\frac{1}{d}$-operations in $l(x)^2$-formula.

- **Circuit-hardness $\Rightarrow$ SOS-hardness**: ($f$ requires size circuit implies $\mathcal{S}(f) \geq s / \log d$); the opposite plausibly doesn’t hold.

- **Koiran–Portier–Tavenas–Thomassé’15**: Newton-polygon-conjecture about roots of similar depth-4 expressions (also here, $l(x)^{\sqrt{d}}$ vs. $d$).

- **Raz’08**: Super-poly-elusive functions eluding degree-2 maps (generic multivariate).
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- [Agrawal-Vinay’08, Gupta-Kamath-Kayal-Saptharishi’13, Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of *unbounded-powers* of multivariates \( \sum \land \omega(1) \sum \Pi \).

- [Koiran’11] Used univariate depth-4 expression of unbounded-powers; also lower bound on the top-fanin (we require SOS-size).
  - SOS-size is neither top-fanin nor the "size" of the depth-4 circuits, rather it is \#\( \sqrt{d} \)-operations in \( \sum \land \Pi \).

- Circuit-hardness \( \Rightarrow \) SOS-hardness (\( f \) requires size circuit implies \( S(f) \geq s/\log d \)); the opposite plausibly doesn’t hold.

- [Koiran-Portier-Tavenas-Thomassé’15] Newton-polygon-conjecture about roots of similar depth-4 expressions (also here, \( l(\sqrt{d}) \) vs. \( d \)).

- [Raz’08] Super-poly-elusive functions eluding degree-2 maps (generic multivariate).
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- [Agrawal-Vinay’08,...,Gupta-Kamath-Kayal-Saptharishi’13,...,Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of *unbounded-powers* of multivariates $\sum \wedge \varpi(1) \sum \prod$.

- [koiran’11] Used univariate depth-4 expression of *unbounded-powers*; also lower bound on the *top-fanin* (we require SOS-size).
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- [Agrawal-Vinay’08,..,Gupta-Kamath-Kayal-Saptharishi’13,..,Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of *unbounded-powers* of multivariates $\sum \land \omega(1) \sum \Pi$.

- [koiran’11] Used univariate depth-4 expression of *unbounded-powers*; also lower bound on the *top-fanin* (we require SOS-size).
  - SOS-size is *neither* top-fanin nor the “size” of the depth-4 circuits, rather it is # $\Pi$-operations in $\sum \land^2 \sum \Pi$-formula.
SOS-hardness and comparison with prior works

SOS-hardness is quite incomparable/weak to previous works:

- [Agrawal-Vinay’08,..,Gupta-Kamath-Kayal-Saptharishi’13,..,Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of unbounded-powers of multivariates \( \sum \land \omega(1) \sum \Pi \).

- [koiran’11] Used univariate depth-4 expression of unbounded-powers; also lower bound on the top-fanin (we require SOS-size).
  
  - SOS-size is neither top-fanin nor the “size” of the depth-4 circuits, rather it is \# \Pi-operations in \( \sum \land ^2 \sum \Pi \)-formula.
  
  - Circuit-hardness \( \implies \) SOS-hardness (\( f \) requires \( s \) size circuit implies \( S(f) \geq s/\log d \)); the opposite plausibly doesn’t hold.

- [Raz’08] Super-poly-elusive functions eluding degree-2 maps (generic multivariate).

- Newton-polygon-conjecture [Koiran’10] and [Koiran-Portier-Tavenas-Thomassé’15] about roots of similar depth-4 expressions (also here, \( l(\sqrt{d}) \) vs. \( d \)).
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- [Agrawal-Vinay’08,..,Gupta-Kamath-Kayal-Saptharishi’13,..,Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of *unbounded-powers* of multivariates $\sum \land^{(1)} \sum \Pi$.

- [koiran’11] Used univariate depth-4 expression of *unbounded-powers*; also lower bound on the *top-fanin* (we require SOS-size).
  - SOS-size is *neither* top-fanin nor the “size” of the depth-4 circuits, rather it is $\# \Pi$-operations in $\sum \land^2 \sum \Pi$-formula.
  - Circuit-hardness $\implies$ SOS-hardness ($f$ requires $s$ size circuit implies $S(f) \geq s/\log d$); the **opposite** plausibly *doesn’t* hold.

- real-$\tau$-conjecture [Koiran’10] and [Koiran-Portier-Tavenas-Thomassé’15] Newton-polygon-$\tau$-conjecture about roots of similar depth-4 expressions (also here, $\omega(\sqrt{d})$ vs. $d$).
SOS-hardness and comparison with prior works

SOS-hardness is quite *incomparable/weak* to previous works:

- [Agrawal-Vinay’08,..,Gupta-Kamath-Kayal-Saptharishi’13,..,Agrawal-Ghosh-Saxena’18] Hardness for special depth-4/3 – sum-of *unbounded-powers* of multivariates $\sum \land \omega(1) \sum \prod$.

- [Koiran’11] Used univariate depth-4 expression of *unbounded-powers*; also lower bound on the *top-fanin* (we require SOS-size).
  - SOS-size is *neither* top-fanin nor the “size" of the depth-4 circuits, rather it is # $\prod$-operations in $\sum \land 2 \sum \prod$-formula.
  - Circuit-hardness $\implies$ SOS-hardness ($f$ requires $s$ size circuit implies $S(f) \geq s/\log d$); the **opposite** plausibly *doesn’t* hold.

- real-$\tau$-conjecture [Koiran’10] and [Koiran-Portier-Tavenas-Thomassé’15] Newton-polygon-$\tau$-conjecture about roots of similar depth-4 expressions (also here, $\omega(\sqrt{d})$ vs. $d$).

- [Raz’08] Super-poly-elusive functions eluding degree-2 maps (generic *multivariate*).
Theorem 1 (Dutta-Saxena-Thierauf'20)

If there exists an SOS-hard polynomial family, then $\text{VP} \neq \text{VNP}$.

Natural analogue of SOS lower bound to hardness of Permanent in the non-commutative settings, [Hrubeš-Wigderson-Yehudayoff'11].

Restrict the degrees of $f_i$ to be $d \cdot o\left(\log d\right)$ and the top-fanin $s = d \cdot o\left(1\right)$.

A stronger SOS-hardness notion with constant $Y$, gives an exponential separation between $\text{VP}$ and $\text{VNP}$. This proof has many technical differences.
Theorem 1 (Dutta-Saxena-Thierauf’20)

If there exists an SOS-hard polynomial family, then \( \text{VP} \neq \text{VNP} \).
Theorem 1 (Dutta-Saxena-Thierauf’20)
If there exists an SOS-hard polynomial family, then VP ≠ VNP.

- Natural analogue of SOS lower bound to hardness of Permanent in the non-commutative settings, [Hrubeš-Wigderson-Yehudayoff’11].
**Theorem 1 (Dutta-Saxena-Thierauf’20)**

If there exists an SOS-hard polynomial family, then $\text{VP} \neq \text{VNP}$.

- Natural analogue of SOS lower bound to hardness of Permanent in the *non-commutative* settings, [Hrubeš-Wigderson-Yehudayoff’11].

- Restrict the degrees of $f_i$ to be $d \cdot o(\log d)$ and the top-fanin $s = d^{o(1)}$. 
**Theorem 1 (Dutta-Saxena-Thierauf’20)**

If there exists an SOS-hard polynomial family, then $\text{VP} \neq \text{VNP}$.

- Natural analogue of SOS lower bound to hardness of Permanent in the *non-commutative* settings, [Hrubeš-Wigderson-Yehudayoff’11].
- Restrict the degrees of $f_i$ to be $d \cdot o(\log d)$ and the top-fanin $s = d^{o(1)}$.
- A stronger SOS-hardness notion with *constant* $\varepsilon$, gives an *exponential* separation between $\text{VP}$ and $\text{VNP}$. This proof has many technical differences.
SOS-hardness and VP vs. VNP
Main Lemma (SOS Decomposition)

Let \( F \) be a field of characteristic \( \neq 2 \). Let \( f(x) \) be an \( n \)-variate polynomial over \( F \) of degree \( d \), computed by a circuit of size \( s \). Then there exist \( f_i \in F[x] \) and \( c_i \in F \) such that

\[
f(x) = s' \prod_{i=1}^{s'} c_i f_i(x)^2,
\]

where \( s' \leq (sd)^{O(\log d)} \), and \( \deg(f_i) \leq \lceil d/2 \rceil \), for all \( i \in [s'] \).

Can we improve \( s' \) to \( \text{poly}((sd)) \)?
Main Lemma (SOS Decomposition)

Let $\mathbb{F}$ be a field of characteristic $\neq 2$. Let $f(\mathbf{x})$ be an $n$-variate polynomial over $\mathbb{F}$ of degree $d$, computed by a circuit of size $s$. Then there exist $f_i \in \mathbb{F}[\mathbf{x}]$ and $c_i \in \mathbb{F}$ such that

$$f(\mathbf{x}) = \sum_{i=1}^{s'} c_i f_i(\mathbf{x})^2,$$

where $s' \leq (sd)^{O(\log d)}$, and $\deg(f_i) \leq \lceil d/2 \rceil$, for all $i \in [s']$. 

Can we improve $s'$ to $\text{poly}(sd)$?
Main Lemma (SOS Decomposition)

Let $\mathbb{F}$ be a field of characteristic $\neq 2$. Let $f(x)$ be an $n$-variate polynomial over $\mathbb{F}$ of degree $d$, computed by a circuit of size $s$. Then there exist $f_i \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$ such that

$$f(x) = \sum_{i=1}^{s'} c_i f_i(x)^2,$$

where $s' \leq (sd)^{O(\log d)}$, and $\deg(f_i) \leq \lceil d/2 \rceil$, for all $i \in [s']$. 

Can we improve $s'$ to poly$(sd)$?
**Algebraic branching programs (ABP).** An ABP is a directed acyclic graph with a starting vertex $s$ with in-degree zero, an end vertex $t$ with out-degree zero. The edge labels are $a_1 x_1 + \ldots + a_n x_n + c \in \mathbb{F}[x]$, where $a_i, c \in \mathbb{F}$.
**Algebraic branching programs (ABP).** An ABP is a directed acyclic graph with a starting vertex $s$ with in-degree zero, an end vertex $t$ with out-degree zero. The edge labels are $a_1x_1 + \ldots + a_nx_n + c \in \mathbb{F}[x]$, where $a_i, c \in \mathbb{F}$.

- The *weight of a path* is the product of labels of the edges in the path.
**Algebraic branching programs (ABP).** An ABP is a directed acyclic graph with a starting vertex $s$ with in-degree zero, an end vertex $t$ with out-degree zero. The edge labels are $a_1 x_1 + \ldots + a_n x_n + c \in \mathbb{F}[x]$, where $a_i, c \in \mathbb{F}$.

- The weight of a path is the product of labels of the edges in the path.
- The polynomial computed by the ABP is the polynomial computed at the end vertex $t$. 
**Algebraic branching programs (ABP).** An ABP is a directed acyclic graph with a *starting vertex* $s$ with in-degree zero, an *end vertex* $t$ with out-degree zero. The edge labels are $a_1x_1 + \ldots + a_nx_n + c \in \mathbb{F}[x]$, where $a_i, c \in \mathbb{F}$.

- The *weight of a path* is the product of labels of the edges in the path.
- The *polynomial computed by the ABP* is the polynomial computed at the end vertex $t$.

This ABP computes

$$x_1x_2x_3 + x_1x_2(1 + x_3) + (1 + x_1)x_2(1 + x_3)$$
Proof idea of Main Lemma

**Proof Sketch.** Here is the basic outline:

1. Wlog, assume it to be a homogeneous function $f$ of degree $d$ computed by size $s$ circuit.
2. Apply result of [Valiant-Skyum-Berkowitz-Rackoff'83] to make it log-depth with $\text{poly}(s)$-size blowup.
3. Convert the circuit to a homogeneous ABP of size (width) $w = s \log d$ such that each edge has linear form weight (without constants).
4. By construction, $i$-th layer nodes compute polynomials of degree exactly $i$.
5. Cut the ABP, at the $d/2$-th layer, we get $f = (f_1, \ldots, f_w)^T \cdot (f_1', \ldots, f_w') = \prod_{i=1}^w f_i \cdot f_i'$, where $f_i$ and $f_i'$ have degree $d/2$. Write each product $f_i \cdot f_i' = \frac{1}{4} \cdot (f_i + f_i')^2 - \frac{1}{4} \cdot (f_i - f_i')^2$, which finally gives the desired decomposition.
Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
- Apply result of [Valiant-Skyum-Berkowitz-Rackoff’83] to make it log-depth with $\text{poly}(s)$-size blowup.
Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
- Apply result of [Valiant-Skyum-Berkowitz-Rackoff’83] to make it log-depth with poly($s$)-size blowup.
- Convert the circuit to a homogeneous ABP of size (width) $w := s^{\log d}$ such that each edge has linear form weight (without constants).
Proof idea of Main Lemma

Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
- Apply result of [Valiant-Skyum-Berkowitz-Rackoff’83] to make it log-depth with poly(s)-size blowup.
- Convert the circuit to a homogeneous ABP of size (width) $w := s^{\log d}$ such that each edge has linear form weight (without constants).
- By construction, $i$-th layer nodes compute polynomials of degree exactly $i$. 
Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
- Apply result of [Valiant-Skyum-Berkowitz-Rackoff’83] to make it log-depth with poly($s$)-size blowup.
- Convert the circuit to a *homogeneous* ABP of size (width) $w := s^\log d$ such that each edge has linear form weight (without constants).
- By construction, $i$-th layer nodes compute polynomials of degree *exactly* $i$.
- *Cut* the ABP, at the $d/2$-th layer, we get
  \[ f = (f_1, \ldots, f_w)^T \cdot \left( f_1', \ldots, f_w' \right) = \sum_{i=1}^{w} f_i \cdot f_i', \]  
  where $f_i$ and $f_i'$ have degree $d/2$. 

Proof idea of Main Lemma

Proof Sketch. Here is the basic outline:

- Wlog, assume it to be a homogeneous $f$ of degree $d$ computed by size $s$ circuit.
- Apply result of [Valiant-Skyum-Berkowitz-Rackoff’83] to make it log-depth with $\text{poly}(s)$-size blowup.
- Convert the circuit to a homogeneous ABP of size (width) $w := s^{\log d}$ such that each edge has linear form weight (without constants).
- By construction, $i$-th layer nodes compute polynomials of degree exactly $i$.
- Cut the ABP, at the $d/2$-th layer, we get $f = (f_1, \ldots, f_w)^T \cdot (f_1', \ldots, f_w') = \sum_{i=1}^w f_i \cdot f_i'$, where $f_i$ and $f_i'$ have degree $d/2$.
- Write each product $f_i \cdot f_i' = 1/4 \cdot (f_i + f_i')^2 - 1/4 \cdot (f_i - f_i')^2$, which finally gives the desired decomposition.
Proof of Theorem 1: SOS-hardness to $VP \neq VNP$

Recall Theorem 1: If an explicit $f_d(x)$ is SOS-hard i.e. $S_{F}(f_d) \geq d^{1/2+\varepsilon}$ for

$\varepsilon = \omega(\sqrt{\log \log d / \log d})$, then $VP \neq VNP$. 

$K$\textit{Wlog, }$f_d$ is SOS-hard with $Y = (\log \log d / \log d)^{1/3}$. $K$Convert this to a $kn$-variate $n$-degree multilinear polynomial $P_n, k$ where $k_n \geq d > (k - 1)^n$, ($n$ and $k$ are both functions of $d$ to be fixed later) and show that the family $\in VNP$, but $\notin VP$.

The conversion is as follows:

- Introduce new variables $y_j, \ell$ where $j \in [n]$ and $\ell \in [0, k - 1]$.
- Monomial $x_i$ in $f_d(x)$ maps to $q(f_d(x)) = \prod_{j=1}^{n} y_j, i_j$, where $i_j = \prod_{j=1}^{n} i_j \cdot k_j - 1$, $0 \leq i_j \leq k - 1$.
- By definition $P_n, k = q(f_d)$ is $kn$-variate $n$-degree multilinear polynomial.

$K$\textit{P}_n, k is very explicit and thus the family $\in VNP$. 

16
Proof of Theorem 1: SOS-hardness to $\text{VP} \neq \text{VNP}$

Recall Theorem 1: If an explicit $f_d(x)$ is SOS-hard i.e. $S_F(f_d) \geq d^{1/2+\varepsilon}$ for $
\varepsilon = \omega(\sqrt{\log \log d / \log d})$, then $\text{VP} \neq \text{VNP}$.

- Wlog, $f_d$ is SOS-hard with $\varepsilon = (\log \log d / \log d)^{1/3}$.
Recall Theorem 1: If an explicit $f_d(x)$ is SOS-hard i.e. $S_{\mathbb{F}}(f_d) \geq d^{1/2+\varepsilon}$ for $\varepsilon = \omega(\sqrt{\log \log d / \log d})$, then VP $\neq$ VNP.

- Wlog, $f_d$ is SOS-hard with $\varepsilon = (\log \log d / \log d)^{1/3}$.

- Convert this to a $kn$-variate $n$-degree multilinear polynomial $P_{n,k}$ where $k^n \geq d > (k - 1)^n$, ($n$ and $k$ are both functions of $d$ to be fixed later) and show that the family $\in$ VNP, but $\notin$ VP.
Recall Theorem 1: If an explicit $f_d(x)$ is SOS-hard i.e. $S_F(f_d) \geq d^{1/2+\varepsilon}$ for $\varepsilon = \omega(\sqrt{\log \log d / \log d})$, then $\text{VP} \neq \text{VNP}$.

- Wlog, $f_d$ is SOS-hard with $\varepsilon = (\log \log d / \log d)^{1/3}$.

- Convert this to a $kn$-variate $n$-degree multilinear polynomial $P_{n,k}$ where $k^n \geq d > (k-1)^n$, ($n$ and $k$ are both functions of $d$ to be fixed later) and show that the family $\in \text{VNP}$, but $\notin \text{VP}$. The conversion is as follows:
Proof of Theorem 1: SOS-hardness to VP ≠ VNP

Recall Theorem 1: If an explicit \( f_d(x) \) is SOS-hard i.e. \( S_{\mathbb{R}}(f_d) \geq d^{1/2+\varepsilon} \) for \( \varepsilon = \omega(\sqrt{\log \log d / \log d}) \), then VP ≠ VNP.

- Wlog, \( f_d \) is SOS-hard with \( \varepsilon = (\log \log d / \log d)^{1/3} \).

- Convert this to a \( kn \)-variate \( n \)-degree multilinear polynomial \( P_{n,k} \) where \( k^n \geq d > (k - 1)^n \), (\( n \) and \( k \) are both functions of \( d \) to be fixed later) and show that the family \( \in \text{VNP} \), but \( \notin \text{VP} \). The conversion is as follows:
  - Introduce new variables \( y_j, \ell \) where \( j \in [n] \) and \( \ell \in [0, k - 1] \).
Proof of Theorem 1: SOS-hardness to VP \( \neq \) VNP

Recall Theorem 1: If an explicit \( f_d(x) \) is SOS-hard i.e. \( S_{\mathbb{F}}(f_d) \geq d^{1/2+\varepsilon} \) for 
\( \varepsilon = \omega(\sqrt{\log \log d / \log d}) \), then VP \( \neq \) VNP.

- Wlog, \( f_d \) is SOS-hard with \( \varepsilon = (\log \log d / \log d)^{1/3} \).

- Convert this to a \( kn \)-variate \( n \)-degree multilinear polynomial \( P_{n,k} \) where 
\( k^n \geq d > (k - 1)^n \), (\( n \) and \( k \) are both functions of \( d \) to be fixed later) and show 
that the family \( \in \) VNP, but \( \notin \) VP. The conversion is as follows:

  > Introduce new variables \( y_{j,\ell} \) where \( j \in [n] \) and \( \ell \in [0, k - 1] \).

  > Monomial \( x^i \) in \( f_d(x) \) maps to \( \phi(x^i) := \prod_{j=1}^{n} y_{j,i_j} \), where 
\( i_j := \sum_{j=1}^{n} i_j \cdot k^{j-1}, \quad 0 \leq i_j \leq k - 1. \)
Recall Theorem 1: If an explicit \( f_d(x) \) is SOS-hard i.e. \( S_{\mathbb{F}}(f_d) \geq d^{1/2+\varepsilon} \) for 
\( \varepsilon = \omega(\sqrt{\log \log d / \log d}) \), then VP \( \neq \) VNP.

\[ \square \quad \text{Wlog, } f_d \text{ is SOS-hard with } \varepsilon = (\log \log d / \log d)^{1/3}. \]

\[ \square \quad \text{Convert this to a } kn\text{-variate } n\text{-degree multilinear polynomial } P_{n,k} \text{ where } \]
\( k^n \geq d > (k - 1)^n \), (\( n \) and \( k \) are both functions of \( d \) to be fixed later) and show 
that the family \( \in \) VNP, but \( \notin \) VP. The conversion is as follows:

\[ \quad \text{Introduce new variables } y_{j,\ell} \text{ where } j \in [n] \text{ and } \ell \in [0, k - 1]. \]

\[ \quad \text{Monomial } x^i \text{ in } f_d(x) \text{ maps to } \phi(x^i) := \prod_{j=1}^n y_{j,i_j}, \text{ where } \]
\[ i =: \sum_{j=1}^n i_j \cdot k^{j-1}, \quad 0 \leq i_j \leq k - 1. \]

\[ \quad \text{By definition } P_{n,k} = \phi(f_d) \text{ is } kn\text{-variate } n\text{-degree multilinear polynomial.} \]
Recall Theorem 1: If an explicit \( f_d(x) \) is SOS-hard i.e. \( S_{\mathbb{F}}(f_d) \geq d^{1/2+\varepsilon} \) for \( \varepsilon = \omega\left(\sqrt{\log \log d / \log d}\right) \), then \( \text{VP} \neq \text{VNP} \).

\( \square \) Wlog, \( f_d \) is SOS-hard with \( \varepsilon = (\log \log d / \log d)^{1/3} \).

\( \square \) Convert this to a \( kn \)-variate \( n \)-degree multilinear polynomial \( P_{n,k} \) where \( k^n \geq d > (k-1)^n \), \( n \) and \( k \) are both functions of \( d \) to be fixed later) and show that the family \( \in \text{VNP} \), but \( \notin \text{VP} \). The conversion is as follows:

\( \Rightarrow \) Introduce new variables \( y_{j,\ell} \) where \( j \in [n] \) and \( \ell \in [0, k-1] \).

\( \Rightarrow \) Monomial \( x_i \) in \( f_d(x) \) maps to \( \phi(x^i) := \prod_{j=1}^{n} y_{j,i_j} \), where
\[
 i =: \sum_{j=1}^{n} i_j \cdot k^{j-1}, \quad 0 \leq i_j \leq k-1.
\]

\( \Rightarrow \) By definition \( P_{n,k} = \phi(f_d) \) is \( kn \)-variate \( n \)-degree multilinear polynomial.

\( \square \) \( P_{n,k} \) is very explicit and thus the family \( \in \text{VNP} \).
Proof of Theorem 1 (continued)

We show that circuit-size $(\mathcal{P}_n, k) = (\mathcal{Q}^l, 1)$ (implying the family $\not\in \text{VP}$).

Proof by contradiction. Suppose $\mathcal{P}_n, k$ has a small-size circuit.

SOS Decomposition shows that $\mathcal{P}_n, k(y) = \sum_{i=1}^{s} c_i \cdot \mathcal{Q}_i(y)^2$, where $\deg(\mathcal{Q}_i) \leq \deg(\mathcal{P}_n, k)/2 \leq n/2$.

Apply $\delta$ both side to get $\delta = \mathcal{P}_n, k = \sum_{i=1}^{s} c_i \cdot \delta(\mathcal{Q}_i)$.

$\delta$ cannot increase the sparsity. Thus, $|\delta(\mathcal{Q}_i)|_0 \leq |\mathcal{Q}_i|_0 \leq (kn + n/2)n/2$.

Hence, $\text{SF}(\delta) \leq s \cdot (kn + n/2)n/2$.

Fix $k, n$ appropriately and show: $s \leq \delta(Y), \text{and } (kn + n/2)n/2 \leq \delta_1/2 + Y/2$.

Thus, $\text{SF}(\delta) \leq \delta(Y) + 1/2 + Y/2 = o(\delta_1/2 + Y)$, a contradiction! □
We show that circuit-size($P_{n,k}$) = $(kn)^{\omega(1)}$ (implying the family $\not\in$ VP).
Proof of Theorem 1 (continued)

- We show that $\text{circuit-size}(P_{n,k}) = (kn)^{\omega(1)}$ (implying the family $\notin VP$).

- Proof by contradiction. Suppose $P_{n,k}$ has a small-size circuit.
We show that circuit-size \( P_{n,k} = (kn)^{\omega(1)} \) (implying the family \( \notin VP \)).

Proof by contradiction. Suppose \( P_{n,k} \) has a small-size circuit.

SOS Decomposition shows that \( P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2 \), where \( \deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2 \).
Proof of Theorem 1 (continued)

- We show that circuit-size($P_{n,k}$) = $(kn)^\omega(1)$ (implying the family $\notin$ VP).
- Proof by contradiction. Suppose $P_{n,k}$ has a small-size circuit.
- SOS Decomposition shows that $P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2$, where $\deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2$.
- Apply $\phi$ both side to get $f_d = \phi(P_{n,k}) = \sum_{i=1}^{s'} c_i \cdot \phi(Q_i)^2$. 

□
Proof of Theorem 1 (continued)

- We show that \( \text{circuit-size}(P_{n,k}) = (kn)^{\omega(1)} \) (implying the family \( \notin \text{VP} \)).

- Proof by contradiction. Suppose \( P_{n,k} \) has a small-size circuit.

- SOS Decomposition shows that \( P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2 \), where
  \[ \deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2. \]

- Apply \( \phi \) both side to get \( f_d = \phi(P_{n,k}) = \sum_{i=1}^{s'} c_i \cdot \phi(Q_i)^2 \).

- \( \phi \) cannot increase the sparsity. Thus, \( |\phi(Q_i)|_0 \leq |Q_i|_0 \leq \binom{kn+n/2}{n/2} \).
Proof of Theorem 1 (continued)

- We show that circuit-size($P_{n,k}$) = $(kn)^{\omega(1)}$ (implying the family $\notin$ VP).

- Proof by contradiction. Suppose $P_{n,k}$ has a small-size circuit.

- SOS Decomposition shows that $P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2$, where $\deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2$.

- Apply $\phi$ both side to get $f_d = \phi(P_{n,k}) = \sum_{i=1}^{s'} c_i \cdot \phi(Q_i)^2$.

- $\phi$ cannot increase the sparsity. Thus, $|\phi(Q_i)|_0 \leq |Q_i|_0 \leq \binom{kn+n/2}{n/2}$.

- Hence, $S_F(f_d) \leq s' \cdot \binom{kn+n/2}{n/2}$. 
Proof of Theorem 1 (continued)

- We show that circuit-size($P_{n,k}$) = $(kn)^{\omega(1)}$ (implying the family $\notin VP$).

- Proof by contradiction. Suppose $P_{n,k}$ has a small-size circuit.

- SOS Decomposition shows that $P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2$, where $\deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2$.

- Apply $\phi$ both side to get $f_d = \phi(P_{n,k}) = \sum_{i=1}^{s'} c_i \cdot \phi(Q_i)^2$.

- $\phi$ cannot increase the sparsity. Thus, $|\phi(Q_i)|_0 \leq |Q_i|_0 \leq (kn+n/2)$.

- Hence, $S_F(f_d) \leq s' \cdot (kn+n/2)$.

- Fix $k,n$ appropriately and show:

$$s' \leq d^{o(\varepsilon)} \text{, and } \binom{kn+n/2}{n/2} \leq d^{1/2+\varepsilon/2}.$$
We show that circuit-size($P_{n,k}$) = $(kn)^{\omega(1)}$ (implying the family $\notin$ VP).

Proof by contradiction. Suppose $P_{n,k}$ has a small-size circuit.

SOS Decomposition shows that $P_{n,k}(y) = \sum_{i=1}^{s'} c_i \cdot Q_i(y)^2$, where $\deg(Q_i) \leq \deg(P_{n,k})/2 \leq n/2$.

Apply $\phi$ both side to get $f_d = \phi(P_{n,k}) = \sum_{i=1}^{s'} c_i \cdot \phi(Q_i)^2$.

$\phi$ cannot increase the sparsity. Thus, $|\phi(Q_i)|_0 \leq |Q_i|_0 \leq \binom{kn}{{n/2}}$.

Hence, $S_F(f_d) \leq s' \cdot \binom{kn+n/2}{{n/2}}$.

Fix $k, n$ appropriately and show:

$s' \leq d^{o(\varepsilon)}$, and $\binom{kn+n/2}{{n/2}} \leq d^{1/2+\varepsilon/2}$.

Thus, $S_F(f_d) \leq d^{o(\varepsilon)+1/2+\varepsilon/2} = o(d^{1/2+\varepsilon})$, a contradiction!
Sum-of-cubes (SOC) model and Blackbox-PIT
Can SOS-hardness give \( \text{PIT} \in \text{P} \)?

Ans: Don't know. Currently the best known is \( \text{QP} \) (when \( Y \) is constant), using result from [KI04].

Can we strengthen the condition/measure to put \( \text{PIT} \in \text{P} \)?

Ans: Yes!

An \( n \)-variate polynomial \( f(x) \in F[x] \) over a field \( F \) is computed as a sum-of-cubes (SOC) if
\[
f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^3,
\]
for some top-fanin \( s \), where \( f_i(x) \in F[x] \) and \( c_i \in F \).

Size of \( f \) in Eqn. (3) is no. of distinct monomials in \( f_i \)’s i.e. \( \sum_{i=1}^{s} \text{supp}(f_i) \).

Eg. \( f(x) = x^3 + 6x^2 = (x+1)^3 - (x-1)^3 + x^3 \). Size of \( f \) in this SOC representation is 2.

Denote the minimal size by support-union \( U_F(f,s) \).
Can SOS-hardness give $\text{PIT} \in \text{P}$?
Blackbox-PIT and Sum-of-cubes (SOC)

- Can SOS-hardness give PIT ∈ P? Ans: Don’t know. Currently the best known is QP (when ε is constant), using result from [KI04].
Can SOS-hardness give PIT ∈ P? Ans: Don’t know. Currently the best known is QP (when ε is constant), using result from [KI04].

Can we strengthen the condition/measure to put PIT ∈ P?
Can SOS-hardness give \( \text{PIT} \in \text{P} \)? Ans: Don’t know. Currently the best known is \( \text{QP} \) (when \( \varepsilon \) is constant), using result from [KI04].

Can we strengthen the condition/measure to put \( \text{PIT} \in \text{P} \)? Ans: Yes!
Blackbox-PIT and Sum-of-cubes (SOC)

- Can SOS-hardness give PIT $\in P$? Ans: Don’t know. Currently the best known is QP (when $\epsilon$ is constant), using result from [KI04].

- Can we strengthen the condition/measure to put PIT $\in P$? Ans: Yes!

- An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a sum-of-cubes (SOC) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^3,$$

(3)

for some top-fanin $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$. 
Can SOS-hardness give PIT $\in P$? Ans: Don’t know. Currently the best known is QP (when $\varepsilon$ is constant), using result from [KI04].

Can we strengthen the condition/measure to put PIT $\in P$? Ans: Yes!

An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a sum-of-cubes (SOC) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^3,$$

for some top-fanin $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

$\Rightarrow$ Size of $f$ in Eqn. (3) is no. of distinct monomials in $f_i$’s i.e. $|\bigcup_{i=1}^{s} \text{supp}(f_i)|$. 

Blackbox-PIT and Sum-of-cubes (SOC)

- Can SOS-hardness give PIT $\in \mathbb{P}$? Ans: Don’t know. Currently the best known is QP (when $\varepsilon$ is constant), using result from [KI04].

- Can we strengthen the condition/measure to put PIT $\in \mathbb{P}$? Ans: Yes!

- An $n$-variate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a sum-of-cubes (SOC) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^3,$$

for some top-fanin $s$, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

- Size of $f$ in Eqn. (3) is no. of distinct monomials in $f_i$’s i.e. $|\bigcup_{i=1}^{s} \text{supp}(f_i)|$.

  Eg. $f(x) := x^3 + 6x^2 = (x + 1)^3 - (x - 1)^3 + x^3$. Size of $f$ in this SOC representation is 2.
Can SOS-hardness give PIT $\in P$? Ans: Don’t know. Currently the best known is QP (when $\varepsilon$ is constant), using result from [KI04].

Can we strengthen the condition/measure to put PIT $\in P$? Ans: Yes!

An $n$-variate polynomial $f(x) \in F[x]$ over a field $F$ is computed as a sum-of-cubes (SOC) if

$$f(x) = \sum_{i=1}^{s} c_i \cdot f_i(x)^3,$$

(3)

for some top-fanin $s$, where $f_i(x) \in F[x]$ and $c_i \in F$.

- Size of $f$ in Eqn. (3) is no. of distinct monomials in $f_i$’s i.e. $|\bigcup_{i=1}^{s} \text{supp}(f_i)|$.

Eg. $f(x) := x^3 + 6x^2 = (x + 1)^3 - (x - 1)^3 + x^3$. Size of $f$ in this SOC representation is 2.

- Denote the minimal size by support-union $U_F(f, s)$. 
SOC-hardness : What to expect

- SOC is a complete model for $\text{char}(\mathbb{F}) \neq 2, 3$ because for any $f(x)$:
  
  $$f = (f + 2)^3/24 + (f - 2)^3/24 - f^3/12.$$
SOC-hardness : What to expect

- SOC is a complete model for char($\mathbb{F}$) $\neq 2, 3$ because for any $f(x)$:
  $$f = \frac{(f + 2)^3}{24} + \frac{(f - 2)^3}{24} - \frac{f^3}{12}.$$  

- Trivially $U_{\mathbb{F}}(f, s) \leq |f|_0 + 1$, for any $s \geq 3$. By counting argument, $U_{\mathbb{F}}(f, s) \geq |f|_0^{1/3}$.  

Seems false over $\mathbb{F} = \mathbb{C}$, $\mathbb{R}$ [dimension argument].

Instead fix $\mathbb{F} = \mathbb{Q}$, [Natural choice for PIT].

[Agrawal'20]: For $s = \Omega\left(\frac{d}{2}\right)$, $U_{\mathbb{Q}}(f, s) = O\left(\frac{d}{2}\right)$; for $s = \Omega\left(\frac{d}{3}\right)$, $U_{\mathbb{Q}}(f, s) = \Theta\left(\frac{d}{3}\right)$.

For $s < o\left(\frac{d}{2}\right)$, we conjecture that most polynomials $f_d$ are SOC-hard.
SOC-hardness: What to expect

- SOC is a complete model for $\text{char}(\mathbb{F}) \neq 2, 3$ because for any $f(x)$:
  $$f = (f + 2)^3/24 + (f - 2)^3/24 - f^3/12.$$  

- Trivially $U_{\mathbb{F}}(f, s) \leq |f|_0 + 1$, for any $s \geq 3$. By counting argument, $U_{\mathbb{F}}(f, s) \geq |f|_{0}^{1/3}$.

- If $|f|_0 \approx d$, then $\Omega(d^{1/3}) \leq U_{\mathbb{F}}(f, s) \leq O(d)$.
SOC-hardness: What to expect

- SOC is a complete model for \text{char}(\mathbb{F}) \neq 2, 3 because for any \( f(x) \):

\[
f = \frac{(f + 2)^3}{24} + \frac{(f - 2)^3}{24} - \frac{f^3}{12}.
\]

- Trivially \( U_{\mathbb{F}}(f, s) \leq |f|_0 + 1 \), for any \( s \geq 3 \). By counting argument, \( U_{\mathbb{F}}(f, s) \geq |f|^{1/3}_0 \).

- If \( |f|_0 \approx d \), then \( \Omega(d^{1/3}) \leq U_{\mathbb{F}}(f, s) \leq O(d) \).

**Definition (SOC-hardness).** A poly\( (d) \)-time explicit univariate polynomial family \((f_d)_d\), where \( f_d \) is of degree \(-d\), is SOC-hard, if there exists a positive constant \( \varepsilon' < 1/2 \) such that \( U_{\mathbb{F}}(f_d, d^{\varepsilon'}) = \Omega(d) \).
SOC-hardness: What to expect

- SOC is a complete model for $\text{char}(\mathbb{F}) \neq 2, 3$ because for any $f(\mathbf{x})$:
  \[ f = \frac{(f + 2)^3}{24} + \frac{(f - 2)^3}{24} - \frac{f^3}{12} . \]

- Trivially $U_\mathbb{F}(f, s) \leq |f|_0 + 1$, for any $s \geq 3$. By counting argument, $U_\mathbb{F}(f, s) \geq |f|^{1/3}_0$.

- If $|f|_0 \approx d$, then $\Omega(d^{1/3}) \leq U_\mathbb{F}(f, s) \leq O(d)$.

**Definition (SOC-hardness).** A poly$(d)$-time explicit univariate polynomial family $(f_d)_d$, where $f_d$ is of degree $-d$, is **SOC-hard**, if there exists a positive constant $\varepsilon' < 1/2$ such that $U_\mathbb{F}\left(f_d, d^{\varepsilon'}\right) = \Omega(d)$.

- Seems false over $\mathbb{F} = \mathbb{C}, \mathbb{R}$ [dimension argument].
SOC-hardness: What to expect

- SOC is a complete model for $\text{char}(F) \neq 2, 3$ because for any $f(x)$:
  \[ f = (f + 2)^3/24 + (f - 2)^3/24 - f^3/12. \]

- Trivially $U_F(f, s) \leq |f|_0 + 1$, for any $s \geq 3$. By counting argument,
  $U_F(f, s) \geq |f|_0^{1/3}$.

- If $|f|_0 \approx d$, then $\Omega(d^{1/3}) \leq U_F(f, s) \leq O(d)$.

Definition (SOC-hardness). A $\text{poly}(d)$-time explicit univariate polynomial family $(f_d)_d$, where $f_d$ is of degree $-d$, is SOC-hard, if there exists a positive constant $\varepsilon' < 1/2$ such that $U_F(f_d, d^{\varepsilon'}) = \Omega(d)$.

- Seems false over $F = \mathbb{C}, \mathbb{R}$ [dimension argument].
- Instead fix $F = \mathbb{Q}$, [Natural choice for PIT].

$x^3 + y^3 = 1$ has no $\mathbb{Q}$ solution.
SOC-hardness: What to expect

- SOC is a complete model for char(F) ≠ 2, 3 because for any f(x):
  \[ f = \frac{(f + 2)^3}{24} + \frac{(f - 2)^3}{24} - \frac{f^3}{12}. \]

- Trivially \( U_F(f, s) \leq |f|_0 + 1 \), for any \( s \geq 3 \). By counting argument, \( U_F(f, s) \geq |f|_0^{1/3} \).

- If \( |f|_0 \approx d \), then \( \Omega(d^{1/3}) \leq U_F(f, s) \leq O(d) \).

**Definition (SOC-hardness).** A poly(d)-time explicit univariate polynomial family \( (f_d)_d \), where \( f_d \) is of degree \(-d\), is **SOC-hard**, if there exists a positive constant \( \varepsilon' < 1/2 \) such that \( U_F(f_d, d^{\varepsilon'}) = \Omega(d) \).

- Seems false over \( \mathbb{F} = \mathbb{C}, \mathbb{R} \) [dimension argument].
- Instead fix \( \mathbb{F} = \mathbb{Q} \), [Natural choice for PIT].
- [Agrawal’20]: For \( s = \Omega(d^{1/2}) \), \( U_Q(f_d, s) = O(d^{1/2}) \); for \( s = \Omega(d^{2/3}) \), \( U_Q(f_d, s) = \Theta(d^{1/3}) \).
SOC-hardness: What to expect

- SOC is a complete model for $\text{char}(\mathbb{F}) \neq 2, 3$ because for any $f(x)$:
  \[ f = \frac{(f + 2)^3}{24} + \frac{(f - 2)^3}{24} - \frac{f^3}{12}. \]

- Trivially $U_F(f, s) \leq |f|_0 + 1$, for any $s \geq 3$. By counting argument, $U_F(f, s) \geq |f|_0^{1/3}$.

- If $|f|_0 \approx d$, then $\Omega(d^{1/3}) \leq U_F(f, s) \leq O(d)$.

**Definition (SOC-hardness).** A poly($d$)-time explicit univariate polynomial family $(f_d)_d$, where $f_d$ is of degree $-d$, is SOC-hard, if there exists a positive constant $\varepsilon' < 1/2$ such that $U_F(f_d, d^{\varepsilon'}) = \Omega(d)$.

- Seems false over $\mathbb{F} = \mathbb{C}, \mathbb{R}$ [dimension argument].
- Instead fix $\mathbb{F} = \mathbb{Q}$, [Natural choice for PIT].
- [Agrawal’20]: For $s = \Omega(d^{1/2})$, $U_Q(f_d, s) = O(d^{1/2})$; for $s = \Omega(d^{2/3})$, $U_Q(f_d, s) = \Theta(d^{1/3})$.
- For $s < o(d^{1/2})$, we conjecture that most polynomials $f_d$ are SOC-hard.
Theorem 2: SOC-hardness to PIT

Proof Idea.
Assume \( f_d \) is SOC-hard for some \( Y' \).

Convert it to \( k = O(1) \)-variate, \( \text{ideg-n} \), \( \text{poly}(n^k) \)-time-explicit polynomial \( P_{n,k} \), using inverse-Kronecker map on \( f_d \) i.e. \( P_{n,k}(x_1, x_2, \ldots, x_{n+k-1}) = f_d \).

Prove that \( (P_{n,k})^n \) is a constant-variate circuit-hard family i.e. \( \text{size}(P_{n,k}) = n \Omega(1) \).

Then, use [Guo-Kumar-Saptharishi-Solomon'19] directly to conclude that PIT \( \in \mathcal{P} \).

Proof by contradiction and use useful SOC Decomposition: Any polynomial \( f \) of degree \( d \) of circuit-size \( s \) can be written as \( f = \sum_{i=1}^{\text{poly}(s,d)} c_i Q^3_i \), where \( \deg(Q^3_i) \leq 4d/11 \). \[ 1/3 < 4/11 < 1/e \]

A binomial counting argument shows that small size of \( P_{n,k} \) implies \( UF(f_d,d,Y') = o(d) \), a contradiction!
**Theorem 2: SOC-hardness to PIT**

**Theorem 2 (Efficient derandomization)**
If there is an SOC-hard polynomial family, then blackbox-PIT ∈ P.

Proof Idea.
Assume \( f_d \) is SOC-hard for some \( Y' \).

Convert it to \( k = O(1) \)-variate, ideg-n, poly(n^k)-time-explicit polynomial \( P_{n,k} \), using inverse-Kronecker map on \( f_d \), i.e., \( P_{n,k}(x_1, x_{n+1}, ..., x_{n^k}) = f_d \).

Prove that \( (P_{n,k})_n \) is a constant-variate circuit-hard family, i.e., \( \text{size}(P_{n,k}) = n \Omega(1) \).

Then, use [Guo-Kumar-Saptharishi-Solomon'19] directly to conclude that PIT ∈ P.

Proof by contradiction and use useful SOC Decomposition: Any polynomial \( f \) of degree \( d \) of circuit-size \( s \) can be written as \( f = \sum_{i=1}^{\text{poly}(s,d)} c_i Q_i \), where \( \deg(Q_i) \leq 4d/11 \). \[ 1/3 < 4/11 < 1/e \]

A binomial counting argument shows that small size of \( P_{n,k} \) implies \( UF(f_d, dY') = o(d) \), a contradiction!
Theorem 2: SOC-hardness to PIT

**Theorem 2 (Efficient derandomization)**
If there is an SOC-hard polynomial family, then blackbox-PIT $\in P$.

**Proof Idea.** Assume $f_d$ is SOC-hard for some $\varepsilon'$.
**Theorem 2: SOC-hardness to PIT**

**Theorem 2 (Efficient derandomization)**

If there is an SOC-hard polynomial family, then blackbox-PIT ∈ P.

**Proof Idea.** Assume $f_d$ is SOC-hard for some $\varepsilon'$.

- Convert it to $k = O(1)$-variate, ideg-$n$, poly($n^k$)-time-explicit polynomial $P_{n,k}$, using inverse-Kronecker map on $f_d$ i.e. $P_{n,k}(x, x^{n+1}, \ldots, x^{(n+1)^{k-1}}) = f_d$. 

A binomial counting argument shows that small size of $P_{n,k}$ implies $UF(f_d, d Y') = o(d)$, a contradiction!
**Theorem 2: SOC-hardness to PIT**

**Theorem 2 (Efficient derandomization)**

If there is an SOC-hard polynomial family, then blackbox-PIT ∈ P.

**Proof Idea.** Assume $f_d$ is SOC-hard for some $\varepsilon'$.

- Convert it to $k = O(1)$-variate, ideg-$n$, poly($n^k$)-time-explicit polynomial $P_{n,k}$, using inverse-Kronecker map on $f_d$ i.e. $P_{n,k}(x, x^{n+1}, \ldots, x^{(n+1)^k-1}) = f_d$.

- Prove that $(P_{n,k})_n$ is a constant-variate circuit-hard family i.e. $\text{size}(P_{n,k}) = n^{\Omega(1)}$. Then, use [Guo-Kumar-Saptharishi-Solomon’19] directly to conclude that PIT ∈ P.
Theorem 2: SOC-hardness to PIT

Theorem 2 (Efficient derandomization)
If there is an SOC-hard polynomial family, then blackbox-PIT ∈ P.

Proof Idea. Assume \( f_d \) is SOC-hard for some \( \varepsilon' \).

- Convert it to \( k = O(1) \)-variate, ideg-\( n \), poly(\( n^k \))-time-explicit polynomial \( P_{n,k} \), using inverse-Kronecker map on \( f_d \) i.e. \( P_{n,k}(x, x^{n+1}, \ldots, x^{(n+1)^{k-1}}) = f_d \).

- Prove that \( (P_{n,k})_n \) is a constant-variate circuit-hard family i.e. \( \text{size}(P_{n,k}) = n^{\Omega(1)} \). Then, use [Guo-Kumar-Saptharishi-Solomon’19] directly to conclude that PIT ∈ P.

- Proof by contradiction and use useful SOC Decomposition: Any polynomial \( f \) of degree \( d \) of circuit-size \( s \) can be written as \( f = \sum_{i=1}^{\text{poly}(s,d)} c_i Q_i^3 \), where \( \deg(Q_i) \leq 4d/11 \). \([1/3 < 4/11 < 1/e]\)
Theorem 2: SOC-hardness to PIT

Theorem 2 (Efficient derandomization)
If there is an SOC-hard polynomial family, then blackbox-PIT $\in P$.

Proof Idea. Assume $f_d$ is SOC-hard for some $\varepsilon'$. 

- Convert it to $k = O(1)$-variate, ideg-$n$, poly$(n^k)$-time-explicit polynomial $P_{n,k}$, using inverse-Kronecker map on $f_d$ i.e. $P_{n,k}(x, x^{n+1}, \ldots, x^{(n+1)^{k-1}}) = f_d$.

- Prove that $(P_{n,k})_n$ is a constant-variate circuit-hard family i.e. $\text{size}(P_{n,k}) = n^{\Omega(1)}$. Then, use [Guo-Kumar-Saptharishi-Solomon’19] directly to conclude that PIT $\in P$.

- Proof by contradiction and use useful SOC Decomposition: Any polynomial $f$ of degree $d$ of circuit-size $s$ can be written as $f = \sum_{i=1}^{\text{poly}(s,d)} c_i Q_i^3$, where $\deg(Q_i) \leq 4d/11$. [$1/3 < 4/11 < 1/e$]

- A binomial counting argument shows that small size of $P_{n,k}$ implies $U_F(f_d, d^{\varepsilon'}) = o(d)$, a contradiction!
Conclusion
Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.
Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.

Prove the existence of a SOS-hard family for the *sum of constantly* many squares.
Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.

Prove the existence of a SOS-hard family for the *sum of constantly* many squares.

Prove the existence of a SOC-hard family for a ‘generic’ polynomial $f$ with rational coefficients ($\mathbb{Q}$).
Conclusion

- Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.

- Prove the existence of a SOS-hard family for the *sum of constantly* many squares.

- Prove the existence of a SOC-hard family for a ‘generic’ polynomial $f$ with rational coefficients ($\mathbb{Q}$).

- Can we optimize $\varepsilon$ in the SOS-hardness condition and prove it for *any* $\omega(\sqrt{d})$?
Conclusion

- Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.

- Prove the existence of a SOS-hard family for the *sum of constantly* many squares.

- Prove the existence of a SOC-hard family for a ‘generic’ polynomial $f$ with rational coefficients ($\mathbb{Q}$).

- Can we optimize $\varepsilon$ in the SOS-hardness condition and prove it for *any* $\omega(\sqrt{d})$? For eg: does proving an SOS lower-bound of $\sqrt{d} \cdot \text{poly}(\log d)$, suffice to show $VP \neq VNP$?
Does the existence of a SOS-hard family solve PIT completely? The current proof technique *fails* to reduce from cubes to squares.

Prove the existence of a SOS-hard family for the *sum of constantly* many squares.

Prove the existence of a SOC-hard family for a ‘generic’ polynomial $f$ with rational coefficients ($\mathbb{Q}$).

Can we optimize $\varepsilon$ in the SOS-hardness condition and prove it for *any* $\omega(\sqrt{d})$? For eg: does proving an SOS lower-bound of $\sqrt{d} \cdot \text{poly}(\log d)$, suffice to show $\text{VP} \neq \text{VNP}$?